UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/083,447	02/26/2002	Kazunori Kishimoto	NEKO 19.481	6047

26304	7590	08/07/2006	

KATTEN MUCHIN ROSENMAN LLP
575 MADISON AVENUE
NEW YORK, NY 10022-2585

EXAMINER
TRIMMINGS, JOHN P

ART UNIT	PAPER NUMBER
2138	

DATE MAILED: 08/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

AUG 0 7 2006

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/083,447
Filing Date: February 26, 2002
Appellant(s): KISHIMOTO, KAZUNORI

Brian E. Hennessey
For Appellant

EXAMINER'S ANSWER (REVISED)

This is in response to the appeal brief filed 2/15/2006 appealing from the Office action mailed 4/29/2005. This is also in response to the BPAI return of the undocketed Appeal to the examiner in order to correct "Section (8) **Evidence Relied Upon,**" of the Examiner's Answer.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is

correct except for new grounds outlined below.

New Grounds of Rejection

Whether or not claims 14-16 are unpatentable under 35 U.S.C. §101 as being

written to non-statutory subject matter.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The following is a listing of the evidence (e.g., patents, publications, Official

Notice, and admitted prior art) relied upon in the rejection of claims under appeal:

The applicant's admitted prior art in the "Background" of the application, and

Figures 5-10 of the application drawings (herein AAPA).

Chen et al., "Test Generation for Crosstalk Induced Delay to Integrated Circuits",

IEEE Test conference Proceedings, 28-30 September 1999, pp 191-200.

5,235,566   MERRILL ET AL.   08-1993

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

Claims 1-4, 6-11 and 13-15 are rejected under 35 U.S.C. §103(a) as being

unpatentable over the applicant's admitted prior art (herein AAPA), where references

are taken from the originally presented Disclosure, "Background of the Invention", pages

1-13 (paragraphs [0001]-[0024]), and drawings FIG.5 through FIG.10 (subsequently

labeled as prior art), and in view of IEEE publication "Test Generation for Crosstalk-

Induced Delay in Integrated Circuits" by Chen et al. (herein Chen).

Claims 5, 12, and 16-18 are rejected under 35 U.S.C. §103(a) as being

unpatentable over the applicant's admitted prior art, where references are taken from

the Disclosure "Background of the Invention", pages 1-13 (paragraphs [0001]-[0024]),

and drawings FIG.5 through FIG.10 (subsequently labeled as prior art), and in view of

IEEE publication "Test Generation for Crosstalk-Induced Delay in Integrated Circuits" by

Chen et al. (herein Chen), and further in view of Merrill, U.S. Patent No. 5235566.

Claims 1, 3-5, 7-12 and 14-16 are independent claims, and are the subject

matter of interest in the appellant's arguments.

The independent claims in the application provide that the propagation delay time on a measurement path is measured by comparing or evaluating the actual value of a register at the end of the measurement path with an expected value. Such a feature is taught in the AAPA in FIG.6-10 and in the discussion of paragraphs [0004]-[0017] (the AAPA).

In particular, <u>claim 1</u> relates to a method for testing a semiconductor integrated circuit (AAPA paragraph [0001]), when a signal for measuring a delay (for example, AAPA FIG.6 flip-flop 10_2 may be set to change the state of the input to flip-flop 10_5 through the combinational circuit as explained in paragraphs [0009] and [0010]), is applied to a measurement path (shown in FIG.6 flip-flop 10_5 begins the "measurement path") on which a delay test is conducted, a signal (for example, Chen, FIG.1 A_{in}, which would be an interfering signal within a comparable combinational circuit such as AAPA FIG.6) having a transition being in phase (Chen FIG.1) or in opposite phase (Chen FIG.2) with the signal for measuring a delay (Chen FIG.1 V_{in}, which is comparable to the output of AAPA FIG.6 flip-flop 10_2) applied to the measurement path is applied to a path that influences crosstalk (Chen FIG.1 A_{in}) to the measurement path (see discussion in Chen page 193 section III). In this manner, a propagation delay time of the signal that propagates through the measurement path under the influence of crosstalk is measured (see AAPA paragraph [0020]). In the method according to claim 1, a propagation delay time of a signal is determined by comparing a value of a flip-flop (for example, AAPA flip-flop 10_8 in FIG.6) receiving the signal outputted from an output end of a

measurement path (AAPA FIG.8 10_7 via measurement path 21, 22, 23) with an expected value (AAPA discussion paragraph [0020]).

Independent <u>claim 3</u> relates to a method for testing a semiconductor integrated circuit in an AC test using a scan path. (AAPA, paragraph [0005]). The method includes receiving from a scan-in terminal of a scan path register (for example, AAPA FIG.8 SCAN IN) a pattern (Chen, page 192, column 1 paragraph 2) for supplying a signal for measuring a delay to a measurement path on which a delay test is conducted and a pattern for supplying a signal having a transition being in phase or in opposite phase with said signal for measuring a delay to a path that influences crosstalk to said measurement path (Chen, page 192 section II discussion on crosstalk). The method of claim 3 also includes supplying said signal for measuring a delay to said measurement path and supplying the signal to the path that influences crosstalk to said measurement path from said scan path register, and reading out a value of the scan path register that samples said signal at an end terminal of said measurement path from a Scan-out terminal (in Chen, page 193 section III, PI's are set to cause delays to clock or data inputs of PO latches in the equivalent AAPA) to compare the value of said scan path register (the equivalent in AAPA is FIG.8 10_8) with an expected value, thereby measuring a delay time in said measurement path (for example, AAPA paragraph [0017]).

The present invention as recited in <u>claim 4</u> relates to a method for testing a semiconductor integrated circuit having a scan path (see below), the measurement object being a clock signal (Merrill, column 1 lines 43-58, Abstract).

A method for testing a semiconductor integrated circuit having a scan path is recited in claim 5 (see below).

A method for testing a semiconductor integrated circuit having a scan path by a computer is recited in claim 7 (see below).

A method for generating patterns for testing a semiconductor integrated circuit having a scan path circuit by a computer is recited in claim 8 (see below).

An apparatus for generating patterns for testing a semiconductor integrated circuit having a scan path circuit is recited in claim 9 (see below).

An apparatus for generating patterns for testing a semiconductor integrated circuit having a scan path circuit is recited in claim 10 (see below).

A method for testing a semiconductor integrated circuit having a scan path circuit as a device under test with an LSI tester is recited in claim 11 (see below).

A method for testing a semiconductor integrated circuit having a scan path circuit as a device under test with an LSI tester as a testing device is recited in claim 12 (see below), the measurement object being a clock signal (Merrill, column 1 lines 43-58, Abstract)..

A computer program product for causing a computer to execute processes for generating patterns for testing a semiconductor circuit having a scan path circuit is recited in claim 14 (see below).

A computer program product for causing a computer to execute processes for generating patterns for testing a semiconductor circuit having a scan path circuit is recited in claim 15 (see below).

A computer program product for causing a computer to execute processes for generating patterns for testing a semiconductor circuit having a scan path circuit is recited in claim 16 (see below), the measurement object being a clock signal (Merrill, column 1 lines 43-58, Abstract).

All of the independent claims 1, 3-5, 7-12 and 14-16 include a feature of "comparing a value of a flip-flop receiving said signal outputted from an output end of said measurement path with an expected value" (claims 1, 7, 8, 9, 10, 14), or similar feature, "comparing the value of said scan path register with an expected value, thereby measuring a delay time in said measurement path" (claim 3), "comparing a value of a flip-flop that samples the signal of an end terminal of said measurement path with an expected value, thereby measuring a delay time in said measurement path" (claims 4 and 5), "comparing the value of the flip-flop that receives the outputted signal outputted fdrom the output end of said measurement path at the data terminal thereof with an expected value" (claims 11 and 12), and "comparing a value of a flip-flop receiving said signal propagated through said measurement path outputted from an output end of said measurement path with an expected value" (claims 15 and 16). These features are discussed in the AAPA, paragraphs [0001]-[0017] and FIG.6-10.

New Grounds of Rejection
The following grounds of rejection are applicable to the appealed claims:
35 U.S.C. §101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 14-16 are rejected under 35 USC §101. The claimed invention in each of

claims 14-16 is directed to non-statutory subject matter. The claims recite, "A computer

program product for causing a computer to execute processes for generating patterns

for testing a semiconductor circuit having a scan path circuit, the program product

comprising the processes of: ...". The limitations of each claim read to a list of activities

per se being executed by a program, but the program as such is not embodied in a

physical medium and therefore is not a physical "thing". The Federal Circuit recognizes

that the fact that a nonstatutory method is carried out on a programmed computer does

not make the process claim statutory. Grams, 888 F.2d at 841, 12 USPQ2d at 1829

(claim 16 ruled nonstatutory even though it was a computer-implemented process).

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions

or expressions of the programs, are not physical "things." They are neither computer

components nor statutory processes, as they are not "acts" being performed. Such

claimed computer programs do not define any structural and functional

interrelationships between the computer program and other claimed elements of a

computer which permit the computer program's functionality to be realized. In contrast,

a claimed computer-readable medium encoded with a computer program is a computer

element which defines structural and functional interrelationships between the computer

program and the rest of the computer which permit the computer program's functionality

to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at

1035. Accordingly, the claims are rejected.

(10) Response to Argument

The applicant's arguments are rebutted herein as follows:

Whether or not claims 1-4, 6-11 and 13-15 are unpatentable under 35 USC §103(a) based on the AAPA in view of the Chen reference.

The appellant argued that there is no motivation to combine the subject references, but the examiner disagrees. The motivation is explicitly found in the reference Chen in page 191, column 2, where Chen states that severe design and test problems caused by crosstalk result in delays in product development and huge expenses. The advantage stated by Chen therein is a method to test for crosstalk that is a cost-effective alternative to expensive re-design of a chip during product development. The method, applied to standard circuit delay measurement methods, provides the manufacturer with a system that detects crosstalk (Chen page 193 section III). One with ordinary skill in the art at the time of the invention, motivated as suggested, would have found it obvious to employ the crosstalk test method of Chen to the AAPA in order to speed product to the customer at lower cost.

The appellant further argued that neither Chen not the AAPA discloses comparing a flip-flop receiving a signal outputted from a measurement path under influence of crosstalk, but the examiner has shown above (see section 9 above, claim 1) that the AAPA and Chen, when combined, disclose the features.

Whether or not claims 5, 12 and 16-18 are unpatentable under 35 USC

§103(a) based on the AAPA in view of the Chen reference, and further in view of

Merrill.

The appellant argued that there is no motivation to combine the subject

references, and that neither Chen not the AAPA discloses comparing a flip-flop

receiving a signal outputted from a measurement path under influence of crosstalk. Both

appellant arguments for these claims are the same as the arguments presented above

in this section for claims 1-4, 6-11 and 13-15. These arguments are both answered by

the examiner using the above answer in this section.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the

Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

This examiner's answer contains a new ground of rejection set forth in section **(9)**

above. Accordingly, appellant must within **TWO MONTHS** from the date of this answer

exercise one of the following two options to avoid *sua sponte* **dismissal of the appeal**

as to the claims subject to the new ground of rejection:

(1) **Reopen prosecution.** Request that prosecution be reopened before the

primary examiner by filing a reply under 37 CFR 1.111 with or without amendment,

affidavit or other evidence. Any amendment, affidavit or other evidence must be

relevant to the new grounds of rejection. A request that complies with 37 CFR

41.39(b)(1) will be entered and considered. Any request that prosecution be reopened

will be treated as a request to withdraw the appeal.

(2) **Maintain appeal.** Request that the appeal be maintained by filing a reply

brief as set forth in 37 CFR 41.41. Such a reply brief must address each new ground of

rejection as set forth in 37 CFR 41.37(c)(1)(vii) and should be in compliance with the

other requirements of 37 CFR 41.37(c). If a reply brief filed pursuant to 37 CFR

41.39(b)(2) is accompanied by any amendment, affidavit or other evidence, it shall be

treated as a request that prosecution be reopened before the primary examiner under

37 CFR 41.39(b)(1).

Extensions of time under 37 CFR 1.136(a) are not applicable to the TWO

MONTH time period set forth above. See 37 CFR 1.136(b) for extensions of time to

reply for patent applications and 37 CFR 1.550(c) for extensions of time to reply for ex
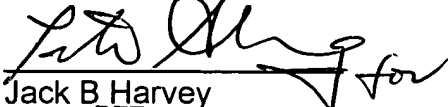
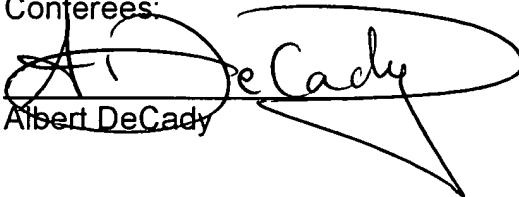parte reexamination proceedings.

Respectfully submitted,

John P Trimmings

A Technology Center Director or designee must personally approve the

new grounds of rejection set forth in section (9) above by signing below:

Jack B Harvey

PETER WONG, DIRECTOR

TECHNOLOGY CENTER 2100

Conferees:

Albert DeCady

ALBERT DECADY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Emmanuel L Moise

Test Generation for Crosstalk-Induced Delay in Integrated Circuits[1]

Wei-Yu Chen, Sandeep K. Gupta and Melvin A. Breuer

Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-2562

Abstract

Due to technology scaling and increasing clock frequency, problems due to noise effects lead to an increase in design/debugging efforts and a decrease in circuit performance. This paper shows how crosstalk coupling between lines can affect the propagation delay of signals in integrated circuits. A model is presented to evaluate the effect of parasitic coupling : crosstalk. Conditions for the creation of the worst-case coupling and propagation of a delayed signal are presented. A test pattern generation algorithm utilizing the above conditions is presented and applied to several example circuits.

I. Introduction

With the scaling of VLSI circuits, the increase in switching speeds, and the mixing of devices with different driving strengths in the same integrated circuit, crosstalk effects are induced between some circuit elements [1 - 4]. These effects can induce faulty behavior and hence become an important problem in testing.

Crosstalk effects can be categorized into two types: crosstalk induced pulses and crosstalk induced delay. The first manifests as a pulse on a line, called the victim line, which should remain in a static state when one or more neighboring lines, called affecting lines, have a transition. Depending on their amplitude and width, these pulses can have an important impact on circuit performance [5]. The second effect, crosstalk delay, is produced when both the affecting and victim lines have simultaneous or near simultaneous transitions. If both lines transit in the same direction, their transition times are reduced, hence the effective delay is said to be reduced. We refer to this phenomenon as crosstalk speedup. If the affecting and victim lines transit in the opposite direction, then there will be an increase in delay, which is called crosstalk slowdown. These unexpected changes in signal propagation delays can cause faulty behaviors. For example, many high performance circuits make extensive use of pipelines, shallow logic blocks between storage elements, dynamic gates, latches instead of flip-flops, single phase clocking, and performance based logic design. The net result is

that the timing margins between clocked elements are small. Hence delay must be well controlled and budgeted. Because crosstalk can adversely affect signal delay, coupling effects must be correctly handled to guarantee correct circuit operation. Thus the delay time for each combinational block and setup and hold times for latch elements must include the signal or clock skews due to crosstalk. Current trends in integrated circuit design indicate that signal noise and skew due to crosstalk can create severe design and test problems. These problems are further aggravated by variations in the fabrication process [22]. If it were not for process variations and stringent area and performance constraints, an error due to crosstalk observed during validation could be eliminated by re-routing signals or redesign. However, redesign may be very expensive in terms of design effort and its impact on a product's schedule. In addition, with process variations and aggressive design goals, it may not always be possible to eliminate all noise effects at all worst case design and fabrication corners. The other alternative is to develop techniques to generate tests for crosstalk. For different process corners a test (or a set of tests) can be generated to detect the presence of errors caused by crosstalk. The resulting tests can be applied to each manufactured chip, and chips in which crosstalk does not cause any error will pass and be shipped to customers, while chips where crosstalk causes an error will be discarded. In other words, a designer can either choose to eliminate potential errors caused by crosstalk via redesign or ignore them until post-manufacturing testing. Such a choice is often made in favor of living with the flaw when there is a time-to-market issue; design changes can be made in a future release. By providing such an alternative, test generation for crosstalk will enable more aggressive design, decrease re-design effort, and/or enable more comprehensive post-manufacturing testing.

Although coupling effects, namely crosstalk, between interconnects have been previously analyzed [5-9], these works focused primarily on the crosstalk-induced pulses and related test pattern generation techniques [3, 10-13]. Crosstalk-induced delay has received less attention. In this paper we first show how the crosstalk effect between signals with concurrent transitions can create a significant increase or decrease in the propagation delay. Then a mixed-signal test

generation process is introduced, where characteristics of crosstalk induced noise are accurately modeled. In addition to traditional logic values, the mixed-signal test generator also includes computation for analog properties such as noise strength (delay) and signal timings (such as arrival time and rise/fall times). New timing conditions are also proposed so that a crosstalk effect, E, is first generated, and then other constraints employed so as to propagate the maximum effect of E, namely delay for slowdown or speedup, or amplitude and width for pulses, to an output or a flip-flop.

The paper is organized as follows. In Section II a model for capacitive coupling is presented to evaluate crosstalk effects with respect to circuit parameters and signal timing properties. In Section III various conditions for the creation of the worst-case coupling and propagation of the crosstalk signal are presented. These conditions are then incorporated into a test generation algorithm to generate tests for crosstalk-induced delay errors. Section IV shows experimental results of the proposed test generation algorithm. Finally, in Section V we present our conclusions.

II. The Crosstalk Effect and Its Characterization

2.1 Crosstalk Effects

Crosstalk is caused by parasitic couplings between adjacent wires that include inductive and capacitive effects. There is a low inductance value that becomes significant at very high frequency in certain lines, such as VDD and GND global buses, which are very long and wide (so R is comparable to ωL) and may conduct large switching current. For signal interconnects the capacitive coupling tends to dominate and it is still feasible to accurately model crosstalk without considering inductance because of the voltage-controlled nature of MOS devices, as detailed in [14]. Fig. 1 shows a simple circuit with coupling capacitance C_m between two signal lines A and V. The value of the parasitic capacitance C_m can be determined as described in [15-17].
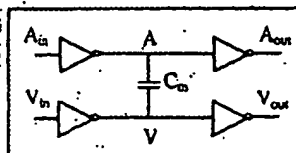

Fig. 1 Simple circuit showing source of crosstalk due to capacitive coupling.

For the purpose of delay calculations the coupling capacitance is commonly modeled as an effective load capacitance. Due to this assumption, the actual signal delay may exceed the amount predicted. For example, consider two coupled lines, as shown in Figure 2. If one line (A) starts a falling transition while the other

(V) is having a rising transition, the waveform on line V may be distorted and the delay increased. The optimism in the commonly used model can be explained in the following way.

Assuming the switching time of the signal on the line A is t_A and that of the signal on the line V is t_V. While the line A is switching, the current going through the coupling capacitance is $C_m(1+ t_V/t_A)V_{DD}/t_V$, for the period of time when both lines are switching. For the remaining portions of the switching time of line V, the coupling capacitance is C_m since the other line finished its switching. Hence there exists a period of time in which the effective coupling capacitance is quite large, namely $C_m(1+ t_V/t_A)$, and for the remaining portion of time is C_m. Therefore the average coupling capacitance is larger than the expected value of C_m and the signal delay is increased.
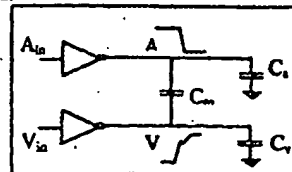

Fig. 2 Two coupled lines.

If two lines are switching in the same direction, by a similar reasoning one can deduce that the signal delay will be decreased. As we scale to deep submicron technologies and multiple layer interconnects, most of the coupling will be between signal lines, and the impact of coupling on signal delay will become increasingly more important and lead to the testing of the crosstalk induced delay.

2.2 A Model for the Evaluation of Crosstalk Effects

To obtain insight into the detailed nature of crosstalk and its dependence on the circuit parameters associated with the coupled lines, consider the lumped model of capacitive coupling shown in Fig. 3.

Fig. 3 Capacitive coupling model.
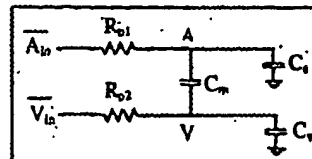
In this model each pulling resistance, R_{p1} and R_{p2}, is composed of the line resistance and the on-channel resistance associated with the line driver, where we assume the complementary device turns off immediately after the transitions are applied at inputs. In [19, 20] it is shown that the impact of neglecting the short circuit current is small provided that the transition

time is short. The load capacitances, C_a and C_v, consist of the line capacitance and the gate capacitance of the load driven by the line. Thus the line driver is equivalent to a pulling resistance, and the coupling network can be viewed as a network of capacitors (C_a, C_v, C_c). This model allows for a somewhat general description of the signals A_{in} and V_{in} not only in terms of their switching rates but also their relative skew. By using Laplace transformations, we can obtain an expression for crosstalk in the s-domain, which we can transform back to the time domain. The analytic response we derive is based on the first order model of MOS device behavior, commonly referred as the LEVEL 1 model, assuming that the channel modulation is negligible. This model was selected because more sophisticated models that take into account higher order effects are intractable for analytic manipulation. The insights gained from the results obtained using this simple model are of value in the development of our ATPG system.

From these analytical expressions, we can analyze effects such as when both signals A and V change simultaneously or with a relative timing skew so that the interference between signal pairs causes distortion on the signal waveforms. Full detail of the derivation of these expressions can be found in [18]. The expressions show that the severity of the crosstalk effect is directly proportional to the coupling capacitance and the ratios of the strengths of the drivers driving the two lines. Also, it was found that the faster the transition at A, the greater the slowdown at V. In addition, the amount of speedup is maximized if the transition on the victim line begins when the pulse induced on this line due to the affecting line reaches its peak. Thus the victim transition lags the affection transition. The amount of crosstalk slowdown is maximum when both signals switch simultaneously. Another important result of the analytic model is that signals on the affecting and victim lines can be skewed up to one or two gate delay and still result in a crosstalk speedup/slowdown effect.

III. Test Generation for Crosstalk-Induced Delay

In this section we present an ATPG algorithm to generate tests for crosstalk delay. We focus on combinational logic circuits whose outputs are either primary outputs or pseudo primary outputs, which are data or clock inputs to storage devices. This algorithm incorporates crosstalk by employing new logic values and corresponding analog information, such as signal arrival times, rise/fall times, and input arrival skews, and searches the space of all possible pairs of input patterns using a significant modified version of backtrace procedure [24]. A signal value in our test generation system contains not only a symbol for its logic value, but also a set of parameters for its corresponding analog properties. For a specific target crosstalk coupling in a

circuit, that can be considered to be analogous to a fault, which we refer to as a c-fault, the objective of this test generator is to generate, under given timing assumptions and requirements, a pair of vectors (a test) that create a crosstalk effect at the target and either a logic error or the maximum noise effect at a primary output. For example, in the case of crosstalk slowdown, given the timing of a clock-edge, the test generator may generate tests that cause a signal to slowdown, and propagate the delayed signal in such a way so as to violate the given timing requirement at a D input to a flip-flop. The symbols and value system shown in Table I are used. The analytic models for the computation of the associated parameters are discussed in [13], and the symbols for pulses will not be discussed since the focus here is on crosstalk delay effects.

Table I Symbols and parameters used for test generation.

Symbols	Associated parameters	Description
1	-	constant 1
0	-	constant 0
P_p	t_a, H, t_H, t_p, t_f	positive pulse
P_n	t_a, H, t_H, t_p, t_f	negative pulse
T_u	t_a, t_r	rising transition
T_d	t_a, t_f	falling transition
S_uT_u	t_a, t_r	speedup rising transition
S_uT_d	t_a, t_f	speedup falling transition
S_dT_u	t_a, t_r	slowdown rising transition
S_dT_d	t_a, t_f	slowdown falling transition
X	-	unknown

Description of parameters.
t_a - arrival time, H, t_H, t_p, t_f as in [13], t_r - rise time, t_f - fall time

3.1 Delay Models and Calculations

To excite a target effect at a specific time (or within a timing window), we first need to obtain some delay information about the gates and paths in the circuit. We associate with each signal line that has a transition a timing window, such as those shown in Fig. 4. The window is defined (bounded) by the minimal arrival ($\tau 1$) and maximal arrival ($\tau 2$) times of the transition. Within this window the transitions with the minimum ($\tau 3$) and maximum ($\tau 4$) transition times can occur in either order, i.e., ($\tau 3$) before ($\tau 4$) or ($\tau 4$) before ($\tau 3$). The window consists of these four transitions. Assuming that the device sizes and output loads of all gates are given, the delay of a gate (assuming one input is switching and other inputs have non-controlling values) can be estimated using standard delay format (SDF) [21]. For a gate g, assuming a timing window is given at an input of the gate, by using SDF and additional computations one can derive the corresponding timing window (four transitions) at the output of the gate. Fig 4. shows an example where input x has a falling transition and output z has a rising transition. Timing windows for various combinations of transitions (rising or falling) at the input and output of the gate can be derived in the same

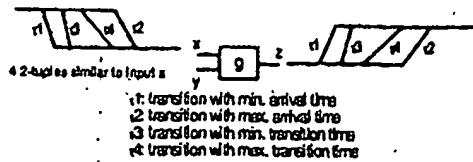manner. Full detail of the derivation for these equations can be found in [23].

4 2-tuples similar to input x

t: transition with min. arrival time
t2 transition with max. arrival time
t3 transition with min. transition time
t4 transition with max. transition time

Fig. 4 Computation of timing windows for a gate.

3.1.1 Forward Delay Calculation

The forward delay calculation actually performs a static timing analysis of the circuit. Given arrival and transition times for signals at primary inputs, one can traverse the circuit starting from PIs in a breadth-first manner to compute timing windows for each line. During the computations for timing windows, in addition to the timing information associated with each line, we also obtained min. and max. input-to-output delays for each input of the gate. For example, for the gate g in Fig. 4 with rising output transitions and falling input transitions at input x, in addition to those four transitions comprising the timing window for each line (input x or output z), there are also two delay values, min. and max. delays (indicating the minimum and maximum delays for propagating a transition from input x to output z), associated with the gate. Similar information is also obtained for input y.

If we would like a signal transition at a specific circuit node to occur near a specific time, these values provide directions and/or choices for a backtrace procedure.

3.1.2 Backward Delay Calculation

The backward delay calculation computes signal required times. Required times are timing windows in which signals are required to appear. Given required times of signals at primary outputs, the calculation process starts at POs and pseudo POs and traverses backward through the circuit to calculate required times by subtracting proper gate delays, which are obtained in the forward delay calculation process. Full detail of the derivation for these equations can also be found in [23].

From these required times, for a given node k, we can find the shortest and longest paths in terms of time to POs by using require times of POs in the fanout cone of node k. In addition, if a signal is to arrive at a specific time at a PO, we can use these parameters to direct the search process in an attempt to satisfy this constraint.

3.2 Timing-Oriented ATPG

With the static timing analysis performed, the timing window of transitions on the affecting line (A)

and victim line (V) can be obtained, as shown in Fig. 5, where a and b (p and q) represents the shortest and longest timing path from the PIs to the affecting (victim) line, and y and x are the shortest and longest timing paths from the victim line to a PO.

Therefore a transition on A can occur no earlier than time a, and no later than time b. Similarly, the timing window for line V is within [p, q]. Also for any signal transition on the victim line to occur in the time interval [T-x, T-y], there is a chance that this signal (on V) will reach a PO at or after the time T, which will result in a possible timing violation, say, a setup time violation. Hence the targeted timing window is the intersected time interval [T-x, q] in which both transitions on A and V can occur to have an effect that may create a timing violation at an output. Note that if this interval [T-x, q] is null, no crosstalk effect for this target can cause a problem at a PO.
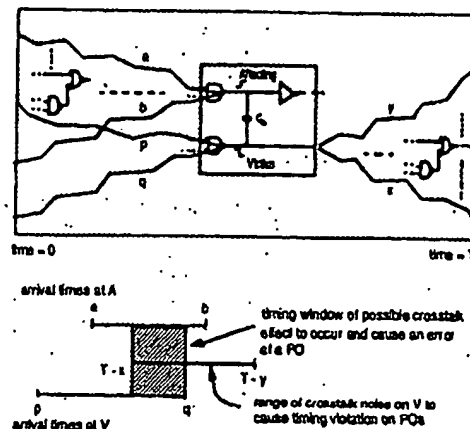

tme = 0   tme = T

Fig. 5 Timing window of transitions on the affecting (A) and victim (V) lines.

3.2.1 Objectives for Crosstalk Delay

There are two types of crosstalk delay: slowdown and speedup. For the rest of this section we will consider only the case of crosstalk slowdown to illustrate the procedures. There are five important objectives in creating a crosstalk slowdown effect of large severity at a specific time.

Objective 1: a late transition on the affecting line
Objective 2: a strong driver on the affecting line.
Objective 3: a transition (opposite direction to that of the affecting line) on the victim line, with a skew bounded by ± α time units
Objective 4: a weak driver on the victim line.
Objective 5: a propagation path that delays the target slowdown-signal as much as possible until it reaches a PO or a flip-flop.

These objectives are used to determine conditions to be satisfied for maximizing the observed crosstalk noise.

Objective 2 and 4 determine conditions that a two-pattern test must satisfy to generate a crosstalk effect of maximal severity, which were presented in [13] using the expressions developed in [18]. Objective 3 provides the maximum acceptable skew between affecting and victim line such that the crosstalk effect is significant. As stated in Section 2.2, the crosstalk slowdown effect is maximum if both the affecting and victim lines switch at the same time, as shown in Fig. 6. We can also see that signals on the affecting and victim lines can be skewed and still result in a crosstalk speedup/slowdown effect. The circuit that is used to derive expressions for Fig. 6 has a typical gate delay of 100ps. If the skew between the affecting and victim line signals is one gate delay and the affecting line signal arrives earlier, then we still have $23/46 = 50\%$ of the maximum crosstalk slowdown effect.
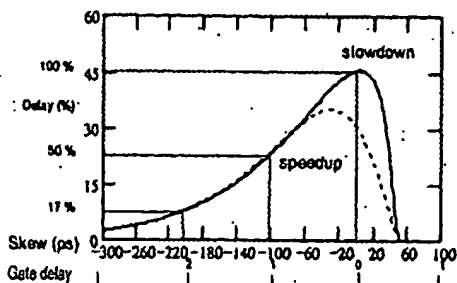

Fig. 6 Amount of speedup and slowdown on the victim line V vs. skew z a negative z implies A leads V.

Although some amount of skew is acceptable to have a significant slowdown effect, initially we attempt to have the victim and affecting lines change at the same time in order to achieve a maximum slowdown effect. If this is not achievable, then a skew, say, up to one gate delay, is allowed for the victim line signal, and we target the arrival time of the victim line signal to be within the window as the arrival time of the affecting line signal plus the allowed skew. Then this timing requirement is translated into the timing-oriented backtrace procedure, described in the next section, to find a test to achieve the desired transition on the victim line.

In summary, together with objective 1 and 5, we try to create on the victim line a late transition, which is slowed down as much as possible due to crosstalk, and propagates through a long path to a PO to cause a timing violation.

3.2.2 Timing-Oriented Backtrace Procedure

In our test generation process, each objective is a 3-tuples in the form of obj(value, timing, condition), where value is the desired signal value, namely a

transition or static value; timing is the target timing requirement which can be specified as earliest, latest, a window or a specific value; and condition is the constraint that specifies whether we want a fast or slow transition, or strong or weak static value.

When an objective is processed, first we check for the existence of a compatible and incomplete pattern at gate inputs. For example, an objective is to have a falling transition at the output of gate g as shown in Fig. 7. We check if the desired timing window $[z_1, z_2]$ overlaps with the timing windows $[A_{tmin}, A_{tmax}]$ that we obtained from the timing analysis described in section 3.1, where A_{tmin}/A_{tmax} denote the minimum/maximum arrival times for falling transitions. If not, then the desired objective cannot be achieved within the desired timing requirement and a new objective must be selected. Next we check whether existing input signals at gate inputs violate the desired output value. For example, as shown in Fig.7, for a desired falling transition at the output, only rising transitions or static 0's are allowed at inputs. If during previous implication process a falling transition has been assigned to one of the inputs, then this objective cannot be achieved and a new objective must be selected.
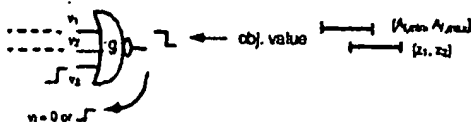

Fig. 7 Check for the existence of a compatible and incomplete pattern at gate inputs in processing objectives.

If the objective seems to be achievable, then for inputs having unknown value "X", we backtrace and search for a pattern to achieve the objective. For the input on which we select to backtrace, we compute the new target timing window $[z_1-d_{max}, z_2-d_{min}]$, where d_{max}/d_{min} are the max/min gate delays from that input to the gate's output under the desired transition (falling transition in this case), as shown in Fig.8. The d_{max}/d_{min} delays are obtained from the static timing analysis described in Section 3.1. Then the new target timing window is inserted into the new objective for the input we selected to backtrace, and we continue the backtrace process recursively.

new target timing window = $[z_1-d_{max}, z_2-d_{min}]$ target timing window = $[z_1, z_2]$
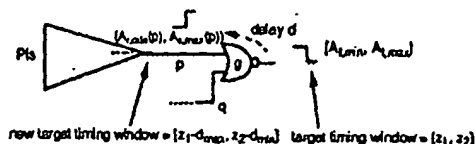
Fig. 8 Recursive execution of the backtrace process.

The third parameter in our objective is the condition that is used for side-fanin assignments. There

are many patterns that can achieve a desired transition on a line with different transition times. For example, to create a falling transition at the output of a two-input NOR gate, both inputs having a rising transition will lead to a shorter gate delay than when one input has a rising transition and the other is held at constant 0. These conditions for side-fan-in assignments that help to create a faster transition were identified in [13]. Table 2 shows conditions for creating a fast transition at the output of a NAND gate. The procedure for side-fan-assignments is similar to the one described in the above subsection.

Table 2 Conditions creating a fast transition at the output of a NAND gate:

Target value at the gate's output.	Necessary Condition ..	Preferred condition on side fan-in	Sufficient condition on side fan-in
T_0	T_0 at one input	All T_0	T_0 or 1
T_0	T_0 at one input	All 1	1 or T_0

However, not only a transition is required, but also the arrival time of the transition is important. It is necessary to check the timing of the side-fan-in assignments so that they won't invalidate the transitions already established. For example, assume that we would like to create a rising transition at the output of a 2-input NAND gate. According to the conditions in [13] we would prefer having both inputs as falling transitions. But if two falling transitions are applied to a 2-input NAND gate, then the earlier transition is the one that controls the timing of the output transition. Hence if we already had a falling transition with the required timing on one input to this NAND gate and the new input has a transition with an earlier transition time, then we may want to discard this transition and try to set this input at a constant "1". Thus, conditions for setting arrival times have higher priority than conditions for switching times.

3.2.3 Dynamic timing refinement

Static timing analysis provides a min-max range for possible transitions on each line. The min-max range is due to unspecified input values. At each ATPG step, as more primary inputs get assigned values, more internal lines have known values and min-max timing ranges shrink due to recalculation of arrival, transition and required times. Hence as we dynamically update the timing information of signals, min/max timing ranges are refined to provide better timing information.

3.2.4 Selection of Propagation Paths

Once the transition signal is created on the victim line and is slowed down by the coupling effect from the affecting line, we want to find a path to further delay the signal as much as possible until it reaches a

PO. There are two situations for slowing down a signal: by side-fan-in assignments and by fan-out branch selections. For example, consider a late (slowdown) rising transition at one input of a 2-input NAND gate. To sensitize the gate to this slowdown signal, the other input of the NAND gate can be either a "1" or a rising transition no later than the slowdown signal. Note that if the side fan-in transition occurs much earlier than the slowdown signal then the side fan-in transition can be regarded as a static "1". However, if the side fan-in transition switches at the same time as the slowdown signal, then the output transition time will be further delayed [13]. The other alternative for a slowdown signal to reach a PO late is to select a longer propagation path. Since we have already performed timing analysis of the circuit as described in Section 3.1.2, the longest delay path from a node to a PO can be easily found. We can also slow down a signal via pulses or hazards on side-fan-ins. Such effects are not dealt with in this paper.

3.3 Conflict between Objectives and Backtracking

In an attempt to achieve the above objectives, there may be some occasions where some decisions made for earlier objectives block the chance of satisfying new objectives. Whenever these situations occur, backtracks are preformed until all objectives are achieved. For example, we may be able to achieve a fast affecting line transition by having all side fan-ins properly assigned, but this may make the desired transition on the victim line impossible to achieve. Hence an immediate backtracking is required to make an adjustment to the PI assignments for creating the affecting line transition so that the victim line transition can be created. Backtracks may affect the quality of the resulting test, for example, they may lead to a stronger victim line driver. The algorithm employed will explore all possible PI combinations so that the best test, if one exists, will be eventually found. So, unlike PODEM which employs a constraint satisfaction search process, our algorithm attempts to maximize an objective, such as maximizing the delay (slowdown) of a signal transition. In the next section we describe a branch and bound technique that evaluates the quality of a partial vector obtained during the TG process, and tries to limit our exploration in the search space.

3.4 Branch and Bound Process to Reduce the Search Space

The order and procedures for processing the objectives provide a way to create a significant crosstalk effect at a PO. They help to find a good solution in an efficient way but they do not guarantee finding an optimum solution. Since there are usually data dependencies between the objectives, it is hard to find an optimum solution without considering more of the circuit's electrical and topological properties. Our

algorithm searches all possible PI assignments to achieve the objectives. To reduce the time complexity of the algorithm we propose to use a branch and bound process.

First, we associate with each gate G a variable $\Omega(G)$. This variable is used to record information about that crosstalk effect that has passed through gate G and has produced the largest amount of slowdown. The initial value of this variable is zero for all gates. Assume the test generation process begins and all the objectives are achieved one by one. When a test vector is found, that is, a crosstalk slowdown signal is propagated to a PO at time T (hence a timing violation occurs), we record information about the crosstalk slowdown signal on all gates along the propagation path starting from the victim line to the PO.

Via backtracking, the test generation continues in order to find a "better" test. If another crosstalk effect reaches gate G, where $\Omega(G) \neq 0$, then we check whether it is possible that this new crosstalk signal will reach a PO later than the recorded Ω value. For a node k in the circuit, the longest path delay from k to a PO reachable from k is ($R_{max} - B_{min}$), where B_{min} is the min. required time of line k and R_{max} is the max. required time of POs reachable from k. Required times are timing windows in which signals are required to appear. Hence we can easily predict the worst case arrival time at a PO as ω = (the arrival time of the crosstalk effect at G + delay of gate G + ($R_{max} - B_{min}$)). If ω is greater than Ω, then we continue because the test constructed may potentially be better than the one we found previously. If not, then we drop this gate from the noise frontier and try another gate in the noise frontier. Thus the Ω variables serve as a bound to limit our search process. If we actually find a new test, then we process the gates along the propagation path and update their Ω values accordingly. We can prune the space more by only considering paths from G to POs that are potentially sensitizable, i.e., have not been blocked by previously assigned PI values.

As the test generation process continues and tests are generated, more gates will be assigned non zero values for the Ω variable and the search space will be further limited. The efficiency is expected to improve significantly by this branch and bound process.

3.5 Test Generation Algorithm

The algorithm consists of five major steps to achieve the objectives. When a test is found, then it is recorded and relevant signal information along the propagation path is stored in the Ω variable to be used for branch-and-bound. Backtrack is performed to explore the search space until all PI combinations have been implicitly tried. The flowchart of the algorithm is shown in Fig. 9.

The outline of the algorithm is as follows.

1) Perform timing analysis of the circuit as described in Section 3.1.
2) Upon START, check the timing window for when transitions on the affecting and victim lines should occur, determine if it is possible to create a timing violation at a PO, and initially target the latest time that both transitions can occur.
3) The initial objectives are to set up desired values on the affecting and victim lines. Conditions and recursive procedures discussed in Section 3.2 are used to guide the timing-oriented backtracing direction so that a weak victim line driver and a fast transition on the affecting line can be satisfied under the desired timing requirements.
4) Once these assignments are made, forward imply and evaluate the actual transition rate on every line. Then create the crosstalk signal on the victim line. The analytic models used for creation of crosstalk effect are presented in [18].
5) We utilize the path delay information from timing analysis of the circuit to select a path to propagate the crosstalk signal to a PO. The path delay is obtained by utilizing required times obtained in the backward delay calculation. We select a path from the current site with the longest path delay for the slowdown case (shortest for the speedup case). When propagating a slowdown signal through a gate, and side fan-ins of the gate are properly assigned to see if they can further delay the crosstalk signal.
6) If the noise effect (crosstalk slowdown) has not reached a PO, then one of the internal signal lines that has a "noise signal" value is used as an objective. If the noise effect reaches a PO, then the PI assignments are recorded as the test. The values of the Ω variables of all gates along the propagation path are updated.
7) Because we desire a test that creates the maximum slowdown at an output, we continue to backtrack so that all possible PI assignments are explored. By the branch and bound process we expect to generate better tests as we continue processing and the search space continues to be reduced in size.
8) Only the signal value 0, 1, T_u, or T_d can be assigned to primary inputs. Whenever a PI is set to a value the implication procedure is performed and the analog timing information, i.e., rise/fall times and/or arrival times, of some signals are re-computed.

The test generation process including objectives, conditions and procedures for crosstalk speedup is similar to that of the crosstalk slowdown process discussed above, except that we want to excite a speedup signal as early as possible and propagate it to a PO through a path with the shortest path delay.

Fig. 9 Flowchart of the algorithm.

IV. Experimental Results

In this section we illustrate the proposed algorithm. Consider the circuit shown in Fig. 10. The channel length of each device is 0.35um and under each gate we indicate the ratio of widths of the PMOS and NMOS FETs. The gain factor ratio of the NMOS to PMOS FETs in the technology file (MOSIS 0.35um) used is 3.5. Assume that the sub-circuit on the left side of the dash line is separated by 1000um from the sub-circuit on the right side of the dash line. Hence lines 14, 24 and 13 are assumed to be about 1000um long and 4 um wide. In addition, assume that line 13 is the affecting line in metal2, line 24 is the victim line in metal3, and they are overlapped so that there is a significant coupling between them (C_m = 280 fF). The gate driving the affecting line 13 is assumed to be a buffer that has a strong driving strength. Gate sizes are computed to achieve signal transition times of 100ps. Primary inputs are assumed to have signal transition times of 100ps. All wire and gate capacitance correspond to a realistic layout, and gate delay is estimated as 110 ps for each gate.
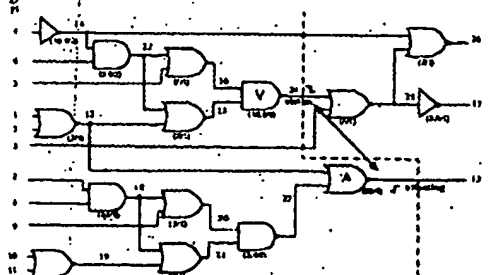

Fig. 10 Example circuit to illustrate the algorithm.

There are six levels of gates from PI to PO. Assume that a 25 ps margin is allowed for each gate and in an aggressive design the clock period is set as (110 + 25) x 6 = 810 ps. Assume that we would like to create a crosstalk slowdown (falling transition) on the victim line 24. First we examine the timing windows for the affecting and victim line transitions. Both the affecting and victim line transitions can occur in time interval [220, 440]. Therefore the transitions for the affecting and victim line are first targeted at 440 ps, i.e., the latest time of the intersection of above time intervals. By backtracing a possible PI assignment can be found where lines 1, 2, 6, and 9 are set to 1; lines 3, 10 set to 0; line 8 set to a rising transition; and lines 4 and 11 set to a falling transition. All conditions for side fan-in assignments are met to have the affecting line transition fast and the victim line transition as slow as possible.

By implication the affecting line starts to transit at time 437 ps with a rise time of 136 ps, and the victim line starts to transit at time 454 ps. We then calculate the crosstalk noise waveform according to the equations in [18]*. The crosstalk slowdown signal on the victim line has an overshoot and a fall time of 370 ps. The waveforms for the affecting and victim line signals are shown in Fig. 11, with the time when the affecting line starts to transit set to zero. Because of the crosstalk effect, there is an increase of 142 ps on the signal delay (50% input to 50% output). This increase is due to both the change in signal slope and the overshoot.

To propagate this slowdown signal, the propagation path has to be sensitized and hence line 5 is set to 0. The slowdown signal propagates to line 17 at a very late time of 811 ps, which violates the timing requirements.

Continuing the test generation process by backtracking in order to explore all PI combinations, we try to find the best test that creates the worst case delay.
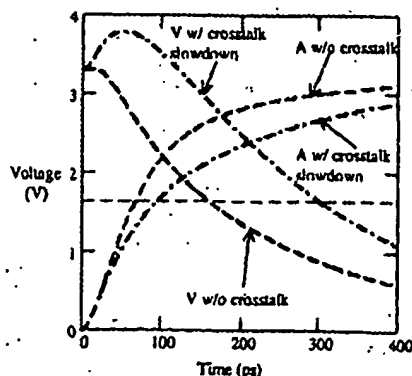

Fig. 11 Waveforms on the victim and affecting lines.

However, since the test we found already creates the fastest affecting line transition and slowest victim line transition under the timing requirements, the branch-and-bound process keeps on pruning the search space and the test we found eventually declares as the worst case vector.

The test generation algorithm described was implemented in the C programming language and applied to several ISCAS '85 benchmark circuits. The program was run on a Pentium II 400 MHz desktop. Since no circuit information, such as crosstalk fault locations, polarity of transitions causing crosstalk fault, coupling capacitance, and layout information, is currently available to us for these circuits, the affecting and victim lines' driver strength and coupling capacitance value are assumed to be sufficient to excite a significant crosstalk noise at a fault site. We assume all transistors are 0.35um in length, the affecting line is driven by a large driver (28um PMOS/8um NMOS), the victim line is driven by a small driver (7um PMOS/2um NMOS), and they run parallel to each other for 1000um distance. All other gates and wires are assumed to have default device sizes and load capacitances.

Two sets of experiments were performed. In the first experiment a single crosstalk delay fault is targeted and the proposed algorithm used to generate all possible tests for the target fault. Tests associated with corresponding crosstalk delay that cause timing violations at POs are recorded so that the test creating the worst case timing violation at a PO can be identified. The results are shown in Table 3. All units are in pico second. The results correlate well with SPICE simulations. The timing criteria in Table 3 is the longest path delay of the circuit plus an extra delay slack of one gate delay. In Table 3 we can see that if there is no crosstalk effect ($C_m = 0$), then there is no timing violation at any primary output. As we increase the coupling capacitance, the victim line signal become more delayed and its transition time increases. Also more test vectors can propagate the crosstalk delay signal to POs. This is because the delay slack is equivalently reduced and some vectors that could not cause timing violation before can do so now. Due to limitation of space, only the result for a small circuit is shown. Similar experiments have been performed on other ISCAS circuits with larger number of nodes.

In the second experiment, for each circuit, 500 pairs of affecting and victim lines are selected at random without considering the circuit structure. The proposed algorithm is applied to generate one test for each fault. Timing criterion and bounding are not used in this experiment. The maximum number of backtracks per fault is limited to 1000. Results of the experiments are shown in Table 4. Column 2 shows the percentage of faults for which tests can be successfully generated. Column 3 shows the number of faults for which an

appropriate test does not exist to propagate a crosstalk delay to a PO and cause a timing violation, and Column 4 shows the number faults for which the number of backtracks exceeds the maximum setting and the TG process was aborted. Column 5 indicates the TG efficiency (Column 2 plus Column 3 divided by 500), and Column 6 is the total CPU time in seconds.

Although in the above experiments the device sizes, coupling capacitance, and related information are artificially inserted, the results in Table 4 demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes, such as a single functional block, within an acceptable amount of time. That is, if all appropriate circuit and layout information is available, our algorithm can identify whether a significant crosstalk fault can be created and propagated to POs and generate an appropriate test.

It is certainly not feasible to apply this algorithm to all pairs of signal lines in a circuit. Only selected pairs of lines should be targeted. The selection of these pairs of lines should be based on the circuit configuration, manufacturing process information, layout, designer's knowledge and other relevant information. This information is typically known in advance to the TG process, and should enable exclusion of some sites that cannot possibly cause errors at outputs.

Table 3. Results of Experiment 1: all tests for a single fault. Circuit C17: affecting node 10 with a rising transition, victim node 16 with a falling transition.

Tests	Arrival time of the crosstalk signal at the fault site (x100ps)	Transition time of the crosstalk signal at the fault site (x100ps)	Arrival time of the of crosstalk signal at a PO (x100ps)
$C_m = 0$ IF 0 test found Timing criteria = 335 No slowdown timing violation at POs			
$C_m = 200$ IF 8 tests found Timing criteria = 335			
$T_d 1 T_d 10$	213	283	335
$T_d 1 T_d 1 T_d$	213	283	335
$T_d T_d 1 T_d 0$	214	287	339
$T_d T_d 1 T_d T_d$	214	287	339
$T_d 1 1 T_d 0$	214	287	339
$T_d 1 1 T_d T_d$	214	287	339
$1 1 T_d 1 0$	214	287	339
$1 1 T_d 1 T_d$	214	287	339
$C_m = 300$ IF 16 tests found Timing criteria = 335			
$T_d T_d T_d T_d 0$: (14 vectors deleted)	251	318	386
$1 1 T_d 1 T_d$	255	324	394

Table 4 Results of Experiment 2: (one test for each fault)
number of faults: 500, no timing criteria.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetectable			
C432	56.8	2.4	40.8	59.2	713
C880	86.6	1.4	12.0	88.0	208
C1355	40.8	0.6	58.6	41.4	2768
C1908	67.6	3.0	29.4	70.6	1680
C2670	66.4	3.6	30.0	70.0	960
C3540	43.6	10	46.4	53.6	3814
C5315	88.8	1.4	9.8	90.2	1008
C7552	74.0	3.0	23.0	77.0	1801

V. Conclusions

Crosstalk effects can have a significant impact on signal delays. To ensure correct circuit operation, coupling effects, such as crosstalk induced signal slowdown and speedup, should have been taken into consideration in estimating the timing of critical paths.

An ATPG algorithm is described to generate tests for crosstalk-induced signal delay, i.e., c-faults. The algorithm models crosstalk delay using new logic values, and utilizes conditions that help excite the maximum crosstalk effect and propagate the crosstalk signal to POs under desired timing requirements. In addition, this ATPG algorithm includes the concept of gate delay, signal arrival time, signal strength and rise/fall times. By using the path delay information obtained by circuit preprocessing, preferred paths can be selected during the backtrace as well as and propagation processes. Because the proposed algorithm implicitly explores all PI combinations, it is necessary to limit the search space to improve efficiency. A branch-and-bound technique is proposed to reduce the search effort. Finally, while most ATPG algorithms attempt to only satisfy a set of logical constraints, this algorithm also maximizes an objective function. Experimental results show that the method can be applied to selected crosstalk faults in circuit of reasonable sizes.

In our TG process the transition on the victim line is created and propagated along a possible long path, and receives the largest amount of crosstalk effect under certain timing requirements. From the point of view of both crosstalk effect and signal propagation, the amount of delay imposed on the victim line signal is maximized with respect to the given constraints. Hence the test vectors generated can be considered as a complementary set of tests for the purpose of delay testing.

Reference
[1] A. K. Goel and Y. R. Huang, "Modeling of crosstalk among the GaAs VLSI connections", IEE Proc. Part G, Vol. 136, pp. 361-368, 1989.
[2] A. Rubio and R. Anglada, "An approach to crosstalk effect analysis and avoidance techniques in digital CMOS VLSI circuits", Int'L Journal of Electronics, Vol. 65, No. 1, pp. 3-17, 1988.

[3] A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.13, No.3, pp.387-394, March 1994.
[4] A. K. Goel, High-speed VLSI Interconnections: Modeling, Analysis, and Simulation, John Wiley & Sons Inc., 1994.
[5] F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis", IEEE Tran. on Circuits and Systems -II: Analog and Digital Signal Processing, Vol. 39, No. 10, pp. 749-752, October 1992.
[6] A. E. Zain and S. Chowdhury, "An analytical method for finding the maximum crosstalk in lossless-coupled transmission lines", Proc. Int'l Conf. on Computed Aided Design, pp. 443-448, 1992.
[7] S. Voranantakul and J. L. Prince, "Crosstalk analysis for high-speed pulse propagation in lossy electrical interconnections", IEEE Trans. on Components, Hybrids, and Manufacturing Technology, Vol. 16, No. 1, pp. 127-136, February 1993.
[8] H. You and M. Soma, "Crosstalk and transient analysis of high-speed interconnects and packages", IEEE Trans. on Solid State Circuits, Vol. 26, pp. 319-30, March 1991.
[9] D. S. Gao, A. T. Yang and S. M. Kang, "Modeling and simulation of interconnection delays and crosstalk in high-speed integrated circuits", IEEE Trans. on Circuits and Systems, Vol. 37, pp.1-9, January 1990.
[10] K. T. Lee, C. Nordquist, and J. A. Abraham "Automatic test pattern generation for crosstalk glitches in digital circuits", Proc. VLSI Test Symposium, pp. 34-39, 1998.
[11] F. Moll and A. Rubio, "Methodology of detection of spurious signals in VLSI circuits", Proc. Europe Test Conference, pp. 491-496, 1993.
[13] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test generation in VLSI circuits for crosstalk noise", Proc. Int'l Test Conf., pp. 641-650, 1998.
[14] R. K. Watts, Submicron Integrated Circuit, New York: Wiley, pp. 317-318, 1989.
[15] Z. Q. Ning, "Capacitance coefficients for VLSI multiple metallization lines", IEEE Trans. on Electron Devices, Vol. ED-34, No. 3, pp. 644-649, March 1987.
[16] N. D. Arora, K.V. Raol and R. Schumann "Modeling and extraction of interconnect capacitance for multilayer VLSI circuits", IEEE Trans. on Computer-Aided Design and Integrated Circuits and Systems, Vol. 15, No. 1, pp. 58-67, January 1996.
[17] N. Delorme, M. Belleville and J. Chilo, "Inductance and capacitance formulas for VLSI interconnects", Electronic Letters, Vol. 32, No. 11, pp. 996-997, May 1996.
[18] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs", Proc. Int'l Test Conf., pp. 809-818, 1997.
[19] N. Hedenstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization", IEEE Trans. on Computer Aided Design, Vol. 6, pp.270-281, March 1987.
[20] A. L. Kayssi, K. A. Sakallah, and T. M. Burks, "Analytical transient response of CMOS inverters", Trans. Briefs, IEEE Trans. on Circuit and Systems, Vol. 39, pp.43-45, January 1992.
[21] IEEE DASC standard delay format (SDF). (See the web page http://vhdl.org/vi/sdf/)
[22] M. A. Breuer and S. K. Gupta, "Process aggravated noise (PAN): new validation and test problems", Proc. Int'l Test Conf., pp. 914-923, 1996.
[23] W. Y. Chen, M. A. Breuer, and S. K. Gupta, "Timing analysis for test generation for crosstalk-induced delay in integrated circuits", Computer Engineering technical report No. 99-04, Electrical Engineering - Systems Department, University of Southern California, April 1999.
[24] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. on Computer, C-30(3), pp. 215-222, 1981.